

Práctica 6. TIPOS DE DATOS ESTRUCTURADOS 1 Tablas.



2 sesiones

Semanas: 5-12 Diciembre



Entrega: 19 Diciembre

OBJETIVOS:

- ✓ Conocer la importancia que tienen las tablas como estructuras de datos en el ámbito de la programación.
- ✓ Conocer los distintos tipos de tablas.
- ✓ Conocer las operaciones posibles a realizar sobre una tabla.
- ✓ Aprender el manejo de tablas bidimensionales.
- ✓ Aprender alguno de los métodos de búsqueda y ordenación más comunes en el tratamiento de información mediante el manejo de estructuras de datos de tipo tabla.

Veamos antes de trabajar con ellas algunas definiciones con las que nos debemos familiarizar:

TABLA .- estructura de datos constituida por un número fijo de elementos, todos ellos del mismo tipo y ubicados en direcciones de memoria físicamente contiguas.

ELEMENTOS DE LA TABLA .- también denominado componente, es cada uno de los datos que forman parte integrante de la tabla.

NOMBRE DE UNA TABLA .- identificativo utilizado para referenciar la tabla y los elementos que la forman.

TIPO DE DATO DE UNA TABLA .- marca el tipo de dato básico que es común a todos y cada uno de los elementos o componentes que forman dicha estructura (numérico, carácter, etc.).

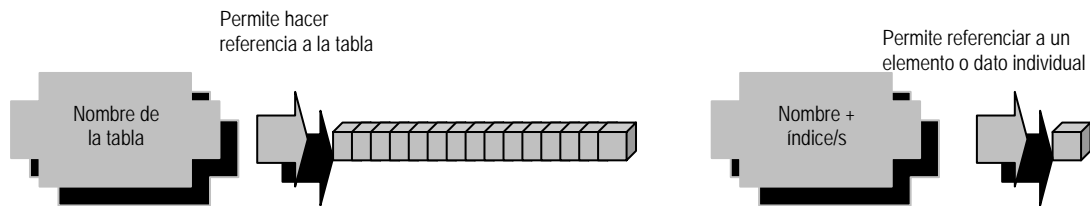
ÍNDICE .- valor numérico entero y positivo a través del cual podemos acceder directa e individualmente a los distintos elementos de una tabla, pues marca la situación relativa de cada elemento o componente dentro de la misma.

TAMAÑO DE UNA TABLA .- el tamaño o longitud de una tabla viene determinado por el número máximo de elementos que la forman, siendo el tamaño mínimo un elemento, y el tamaño máximo n elementos.

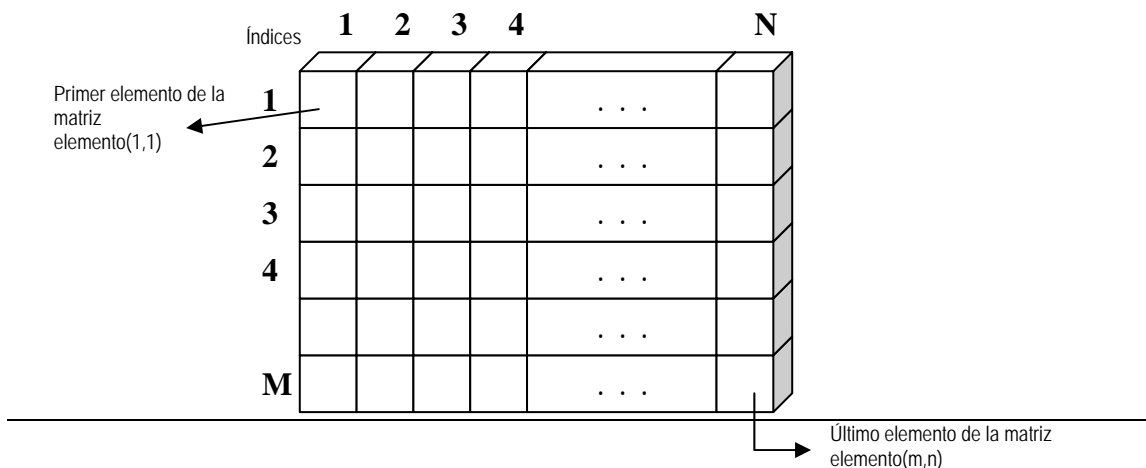
ACCESO A LOS ELEMENTOS O COMPONENTES DE UNA TABLA (INDICE) .- los elementos o componentes de una tabla tratados individualmente son auténticos datos básicos que reciben el mismo trato que cualquier otra variable, con un tipo de dato que coincide con el tipo de la tabla y una denominación propia que les distingue del resto de elementos o componentes que constituyen dicha estructura. Para acceder o referenciar un elemento en particular es suficiente con indicar el nombre de la tabla seguido del índice (posición relativa que ocupa dicho elemento dentro de la tabla) correspondiente a dicho elemento entre paréntesis.

DIMENSIÓN DE LA TABLA .- viene determinado por el número de índices que necesitamos para acceder a cualquiera de los elementos que la forman.

Existen tablas de una dimensión (unidimensionales).



Y tablas de más de una dimensión. Como ejemplo de tabla n-dimensional podemos ver una tabla de 2 dimensiones. En este caso particular de 2 dimensiones se le denomina matriz o tabla bidimensional.



Ejercicio Resuelto 1. Indica para la siguiente tabla cuales son:

'E'	'F'	'M'	'A'	'M'	'J'	'J'	'A'	'S'	'O'	'N'	'D'	Tabla: meses
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	--------------

- nombre de la tabla: *meses*
- tipo: *carácter*
- elementos de la tabla: *12*
- índices: *1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 ó bien 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 ó ...*
- rango o dimensión: *12*

Ejercicio Resuelto 2. Indica cuales son las operaciones básicas que se pueden realizar sobre una tabla.

Las operaciones básicas son:

- *lectura de la tabla*
- *escritura en la tabla*
- *recorrido de la tabla o acceso secuencial*
- *inicialización de la tabla*
- *actualización de la tabla*
- *búsqueda de un elemento en la tabla*
- *ordenación de la tabla*

Ejercicio Resuelto 3. Realiza un módulo (acción) en pseudocódigo que lea una cadena de caracteres y una vez escrita incorpórala al siguiente algoritmo.

algoritmo: ejercicio_caracteres
accion leecadena(...)

...

faccion

escribe('Escribe una palabra y finaliza con un .');
leecadena(...)

falgoritmo.

Solución:

```
algoritmo: ejercicio_caracteres
  const ktam := 20 fconst

  # declaración de un nuevo tipo Tcadena
  tipo Tcadena = tabla [1..ktam] de caracter ftipo

  var frase : Tcadena fvar

  # acción que leerá un vector de caracteres de golpe
  accion leecadena (var vector : Tcadena)
    var contador : entero fvar

    contador := 0;
    repetir
      contador := contador + 1;
      lee(vector[contador])
    hasta ((contador>=ktam) o (vector[contador]='.')) frepetir;
    leelin()
  faccion

  # acción que permite visualizar los datos almacenados en un
  # vector
  accion escribecadena (vector : Tcadena)
    var contador : entero fvar

    contador := 1;
    mientras ((contador <= ktam) y (vector[contador]<>'.')) hacer
      escribe(vector[contador]);
      contador := contador + 1
    fmientras;
    escribelin()
  faccion

  # algoritmo principal que llama a los módulos de arriba
  escribe('Introduce una cadena finalizada con punto (.) :');
  leecadena(frase);
  escribelin('ESTA ES LA CADENA QUE HAS ESCRITO : ');
  escribecadena(frase)
falgoritmo.
```

Ejercicio Resuelto 4. Implementa el algoritmo anterior en lenguaje C.

Solución :

```
// incluir funciones predefinidas de Entrada/Salida
#include <iostream.h>

// declaración de la constante que establece el tamaño máximo de caracteres de nuestras cadenas
#define KTAM 20

// declaración de un nuevo tipo Tcadena
typedef char Tcadena[KTAM];

// declaración de funciones de usuario
void LeeCadena (Tcadena vector);
void EscribeCadena (Tcadena vector);

// cuerpo principal del algoritmo ejercicio_caracteres
main()
{
    Tcadena frase;

    cout << "Introduce una cadena finalizada con punto (.) :";
    LeeCadena(frase);
    cout << "ESTA ES LA CADENA QUE HAS ESCRITO : ";
    EscribeCadena(frase);
}

// función que guardará en un vector una cadena (secuencia de caracteres) leída carácter a carácter
void LeeCadena (Tcadena vector)
{
    // declaración de variables locales
    int contador;

    contador = -1;
    do {
        contador = contador + 1;
        vector[contador] = cin.get();
    } while ( (contador < KTAM) && (vector[contador] != '.'));

    // en C, el carácter de terminación de las cadenas es '\0'
    vector[contador] = '\0';

    // leer el salto de línea
    cin.get();
}

// función que permite visualizar todos los datos almacenados en un vector
void EscribeCadena (Tcadena vector)
{
    // declaración de variables locales
    int contador;

    contador = 0;

    while ( (contador <= KTAM) && (vector[contador] != '\0') ) {
        // imprimir un carácter de vector
        cout << vector[contador];

        contador = contador + 1;
    }

    // Imprime salto de línea
    cout << endl;
}
```

Ejercicio Resuelto 5. ¿Por qué crees que pedimos al usuario que finalice la palabra con un símbolo (el punto en este caso) ?

Para utilizarlo como una marca que determina hasta qué elemento de la tabla (array) se ha inicializado, es decir, es una marca de finalización de los datos que contiene la tabla, de forma que no estamos obligados a utilizar todos sus elementos a la hora de recorrerlo, de ordenarlo, etc.

Ejercicio Obligatorio 1. Implementar un programa en C que dada una frase imprima las palabras de la misma que tengan al menos una vocal repetida. El programa deberá leer una frase que se introduzca por teclado e imprimir por pantalla las palabras que cumplan la condición antes mencionada.

Ejemplo: Si se introduce la siguiente frase :

“Buscar vocales repetidas en las palabras de cualquier frase”

El resultado debería ser la impresión por pantalla de las siguientes palabras:

repetidas
palabras
cualquier

ya que en estas palabras se repite al menos una vocal.

Nota : Por simplicidad, suponed que las palabras de la frase están separadas por un espacio en blanco.

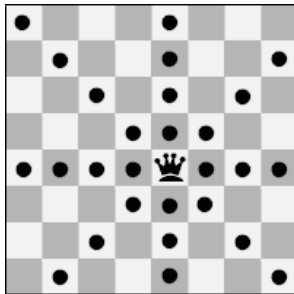
Ejercicio Voluntario 1. Basándose en el ejercicio anterior, añadir un procedimiento al programa implementado que dada la frase introducida por teclado, imprima por pantalla las palabras que cumplan la misma condición indicada en dicho ejercicio (tener al menos una vocal repetida) pero con la restricción de que la repetición de una vocal se consiga sin vocales intermedias distintas.

Ejemplo: Dada la misma frase del ejercicio anterior, el resultado debería ser la impresión por pantalla de las siguientes palabras:

repetidas
palabras

la palabra “cualquier” ahora ya no se imprime porque entre las ocurrencias de la vocal “u” que se repite, aparece otra vocal distinta entre dichas ocurrencias (“a”)

Ejercicio Obligatorio 2. Se trata de implementar un programa en C que simule un tablero de ajedrez y que sea capaz de determinar las posiciones a las que se puede mover la Reina dada una posición inicial de ésta.



Por simplicidad, se considerará que la Reina es la única pieza que hay en el tablero.

La ejecución del programa deberá seguir los siguientes pasos:

- 1.- Mostrar tablero
- 2.- Elegir posición inicial
- 3.- Mostrar posiciones posibles
- 4.- Elegir posición final
- 5.- Mostrar posición final

A continuación, se describe cada paso y se muestra un ejemplo de ejecución:

1.MOSTRAR TABLERO

Se visualizará lo siguiente:

	A	B	C	D	E	F	G	H
1
2
3
4
5
6
7
8

Los puntos indican posición vacía, y las posiciones se referencian con un índice para las filas (de 1 a 8) y otro para las columnas (de 'A' a 'H')

2.ELEGIR POSICIÓN INICIAL

Se solicita por pantalla la introducción de una posición inicial de la Reina (columna, fila), validando que es correcta según los rangos indicados anteriormente.:

Introduce la posición inicial: E5

Y se deberá mostrar esta posición (indicada con el carácter 'i') en el tablero:

A	B	C	D	E	F	G	H
1
2
3
4
5	.	.	.	i	.	.	.
6
7
8

3.MOSTRAR POSICIONES POSIBLES

El programa debe determinar las posiciones posibles a las que se puede mover la reina desde la posición inicial indicada en el paso anterior, y mostrarlas de la siguiente forma:

	A	B	C	D	E	F	G	H
1	?	.	.	.	?	.	.	.
2	.	?	.	.	?	.	.	?
3	.	.	?	.	?	.	?	.
4	.	.	.	?	?	?	.	.
5	?	?	?	?	i	?	?	?
6	.	.	.	?	?	?	.	.
7	.	.	?	.	?	.	?	.
8	.	?	.	.	?	.	.	?

Donde el carácter ‘?’ denota una posición posible

4.ELEGIR POSICIÓN FINAL

Se solicita por pantalla la introducción de la posición final de la Reina (columna, fila), validando que es correcta según las posiciones posibles calculadas en el paso anterior:

Introduce la posición final: A1

5.MOSTRAR POSICIÓN FINAL

Se debe mostrar el tablero completo con la posición inicial (i) y la posición final (f)

	A	B	C	D	E	F	G	H
1	f
2
3
4
5	.	.	.	i
6
7
8

Ejercicio Voluntario 2. Modificar el último paso del ejercicio anterior de forma que se muestre el camino seguido por el movimiento de la Reina seleccionado.

Según el ejemplo anterior, el último paso debería mostrar lo siguiente:

A	B	C	D	E	F	G	H
1	f
2	.	x
3	.	.	x
4
5	.	.	.	i	.	.	.
6
7
8

Donde el carácter ‘x’ denota el camino seguido desde la posición inicial ‘i’ a la posición final ‘f’